|  |
| --- |
| Univerzita Pardubice – Fakulta elektrotechniky a informatiky |
| Semestrální práce z DAS2 a WWW |
| Databázová část |

|  |
| --- |
| Matěj Trakal  8.12.2009 |

# Obsah

[Kapitola 1: Obsah 2](#_Toc248596577)

[Kapitola 2: Základní charakteristika 4](#_Toc248596578)

[Podkapitola 1: Zvolené téma 4](#_Toc248596579)

[Podkapitola 2: Určení funkce databáze 4](#_Toc248596580)

[Podkapitola 3: UML use case diagram 4](#_Toc248596581)

[Kapitola 3: Základní vlastnosti databáze 5](#_Toc248596582)

[Podkapitola 1: Ošetření vstupních dat – SQL Injection 5](#_Toc248596583)

[Podkapitola 2: Integritní omezení 5](#_Toc248596584)

[Podkapitola 3: Propojení databáze s webovou částí 5](#_Toc248596585)

[Navázání spojení s databází 6](#_Toc248596586)

[Vykonání vybraných SQL dotazů pomocí PHP 6](#_Toc248596587)

[Odpojení od databáze 7](#_Toc248596588)

[Kapitola 4: E-R Diagram 8](#_Toc248596589)

[Kapitola 5: Popis tabulek a jejich atributů 10](#_Toc248596590)

[TZanr 10](#_Toc248596591)

[TTitulZanr 10](#_Toc248596592)

[TFormat\_filmu 10](#_Toc248596593)

[TFormat\_titulku 10](#_Toc248596594)

[TJazyk 10](#_Toc248596595)

[TTitulky 10](#_Toc248596596)

[TJazyk\_filmu 10](#_Toc248596597)

[TRole 10](#_Toc248596598)

[TUzivatel 10](#_Toc248596599)

[TTitul 11](#_Toc248596600)

[TAdresa 11](#_Toc248596601)

[TPujcujici 11](#_Toc248596602)

[TVypujcka 11](#_Toc248596603)

[TFilmoteka 11](#_Toc248596604)

[Kapitola 6: Popis a syntaxe použitých databázových objektů 12](#_Toc248596605)

[Podkapitola 1: Funkce 12](#_Toc248596606)

[Podkapitola 2: Trigery 12](#_Toc248596607)

[Podkapitola 3: Indexy 13](#_Toc248596608)

[Podkapitola 4: Sekvence 13](#_Toc248596609)

[Podkapitola 5: Pohledy 13](#_Toc248596610)

[Podkapitola 6: Defaultní předpřipravená data pro fungování webové části 13](#_Toc248596611)

[Podkapitola 7: Další operace nad databází 14](#_Toc248596612)

[Kapitola 7: Závěr 15](#_Toc248596613)

[Kapitola 8: Seznam obrázků 16](#_Toc248596614)

# Základní charakteristika

## Zvolené téma

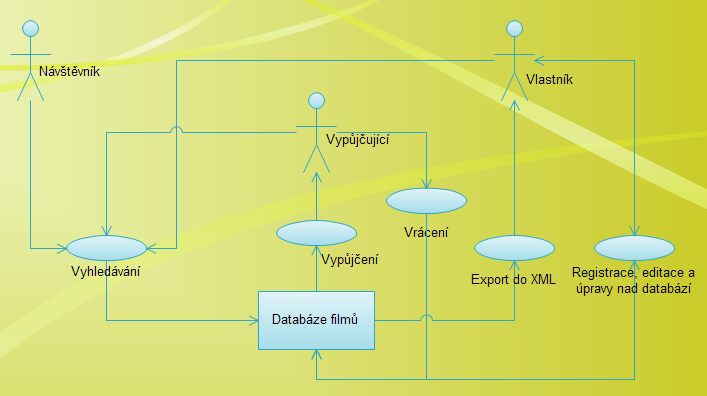
Pro semestrální práci jsem si zvolil tvorbu filmové databáze v plánovaném rozsahu asi 5 tabulek. Bohužel rozsah se výrazně zvětšil postupným zjišťováním návazností a potřeb dělení a zachování základních norem.

## Určení funkce databáze

Databáze má sloužit pro jednoho uživatele vlastnícího domácí filmotéku. Vlastník je hodný a tedy umožňuje svým známým výpůjčku jeho filmů, Je tedy do jisté míry implementována i půjčovna.

Jelikož vlastník využívá více fyzických úložišť filmů a filmy má často i vícekrát v různém formátu a jazycích, musí databáze pojmout i tyto požadavky. Z tohoto hlediska byla ve výsledku databáze navržena jako robustnější a odděluje od sebe jednotlivé tituly a nadále pak jednotlivé filmy umístěné na nějakém médiu.

## UML use case diagram



Obrázek UML Use case diagram

# Základní vlastnosti databáze

## Ošetření vstupních dat – SQL Injection

Díky využití kvalitního databázového layeru (frameworku) Dibi (http://dibiphp.com/cs/) je zajištěno, že provádění sql příkazů nemůže už z principu fungování Dibi být napadeno pomocí SQL Injection. Jediná nevýhoda z jejího nasazení je nutnost mít na serveru PHP ve verzi novější, než 5.2.0 (optimálně 5.3 a vyšší), jelikož využívá pokročilých funkcí PHP pro svůj chod.

## Integritní omezení

***ALTER TABLE*** *TFILMOTEKA* ***ADD******FOREIGN KEY*** *(FILM\_ID)* ***REFERENCES*** *TTITUL (FILM\_ID)  
/****ALTER TABLE*** *TTITULZANR* ***ADD******FOREIGN KEY*** *(FILM\_ID)* ***REFERENCES*** *TTITUL (FILM\_ID)  
/****ALTER TABLE*** *TFILMOTEKA* ***ADD FOREIGN KEY*** *(FORMATFILMU\_ID)* ***REFERENCES*** *TFORMAT\_FILMU (FORMATFILMU\_ID)  
/****ALTER TABLE*** *TVYPUJCKA* ***ADD FOREIGN KEY*** *(KATALOGOVE\_CISLO)* ***REFERENCES*** *TFILMOTEKA (KATALOGOVE\_CISLO)  
/****ALTER TABLE*** *TTITULKY ADD* ***FOREIGN KEY*** *(KATALOGOVE\_CISLO)* ***REFERENCES*** *TFILMOTEKA (KATALOGOVE\_CISLO)  
/****ALTER TABLE*** *TJAZYK\_FILMU* ***ADD FOREIGN KEY*** *(KATALOGOVE\_CISLO)* ***REFERENCES*** *TFILMOTEKA (KATALOGOVE\_CISLO)  
/****ALTER TABLE*** *TVYPUJCKA* ***ADD FOREIGN KEY*** *(NICK)* ***REFERENCES*** *TPUJCUJICI (NICK)  
/****ALTER TABLE*** *TPUJCUJICI* ***ADD******FOREIGN KEY*** *(ADRESA\_ID)* ***REFERENCES*** *TADRESA (ADRESA\_ID)  
/****ALTER TABLE*** *TJAZYK\_FILMU* ***ADD FOREIGN KEY*** *(JAZYK\_ID)* ***REFERENCES*** *TJAZYK (JAZYK\_ID)  
/****ALTER TABLE*** *TTITULKY* ***ADD******FOREIGN KEY*** *(JAZYK\_ID)* ***REFERENCES*** *TJAZYK (JAZYK\_ID)  
/****ALTER TABLE*** *TTITULKY* ***ADD FOREIGN KEY*** *(FORMATTITULKU\_ID)* ***REFERENCES*** *TFORMAT\_TITULKU (FORMATTITULKU\_ID)  
/****ALTER TABLE*** *TUZIVATEL* ***ADD FOREIGN KEY*** *(ROLE\_ID)* ***REFERENCES*** *TROLE (ROLE\_ID)  
/****ALTER TABLE*** *TPUJCUJICI* ***ADD FOREIGN KEY*** *(NICK)* ***REFERENCES*** *TUZIVATEL (NICK)  
/*

## Propojení databáze s webovou částí

Díky tomu, že jsem využil do jisté míry výhod OOP a databázového layeru Dibi se mi práce s databází značně ztížila, ač přinesla i spoustu výhod. Hlavní nevýhodou je nutnost zapouzdření, dále striktní oddělení datové, aplikační a prezentační vrstvy. Jako výhodu jsem naproti tomu získal bezpečnost a modifikovatelnost aplikace.

### Navázání spojení s databází

public function \_\_construct() {  
 try {  
 dibi::connect(array(  
 'driver' => 'oracle',  
 'database' => 'sql101.upceucebny.cz:1521/oracle10',  
 'username' => 'st22312',  
 'password' => HESLO, /\* globální proměnná načítaná z externího souboru ve složce nepřístupné z webového prohlížeče \*/  
 'charset' => 'UTF8'  
 ));  
 } catch (DibiException $e) {  
 echo get\_class($e), ': ', $e->getMessage(), "\n";  
 }  
 }

### Vykonání vybraných SQL dotazů pomocí PHP

/\*\*  
 \* Ověření uživatele skrz databázi  
 \* @param <string> $username  
 \* @param <string> $password  
 \* @return <string> Array[][]  
 \*/  
 public function UserAuth($username, $password) {  
 $result = null;  
 if(dibi::isConnected()) {  
 $result= dibi::query("select role as ROLE, nick as USERNAME, password as PASSWORD from TUZIVATEL left join TROLE on TROLE.ROLE\_ID=TUZIVATEL.ROLE\_ID where NICK='".$username."' and PASSWORD='".$password."'");  
 return $result->fetchAll();  
 }  
 }

/\*\*

\* Vrací informace o titulech - parsované jako XML

\* @return <Array['XML']> parsované XML

\*/

public function MovieParseXML() {

$result = null;

$row = array();

if(dibi::isConnected()) {

$result= dibi::query("select xmlelement(`titul`, xmlforest(film\_id as `id-filmu`,cz as `nazev-cesky`,en as `nazev-anglicky`,original as `nazev-originalni`,delka as `delka`,rok\_vydani as `datum-vydani`,csfd as `odkaz-csfd`,imdb as `odkaz-imdb`,popis as `popis-filmu`)) as xml from ttitul order by cz");

$row = $result->fetchAll();

return $row;

}

}

/\*\*

\* Doplňování názvu filmů

\* @param <string> $var

\* @return <Array[][]> vícerozměrné pole

\*/

public function Autocomplete($var) {

$result = null;

$row = array();

if(dibi::isConnected()) {

$result= dibi::query("select film\_id as IDFILMU, cz as CZ, en as EN, original as ORIGINAL, to\_char(substr(popis,0,50)) as POPIS from ttitul where lower(ttitul.cz) like lower('%".$var."%') or lower(ttitul.en) like lower('%".$var."%') or lower(ttitul.original) like lower('%".$var."%')");

$row = $result->fetchAll();

return $row;

}

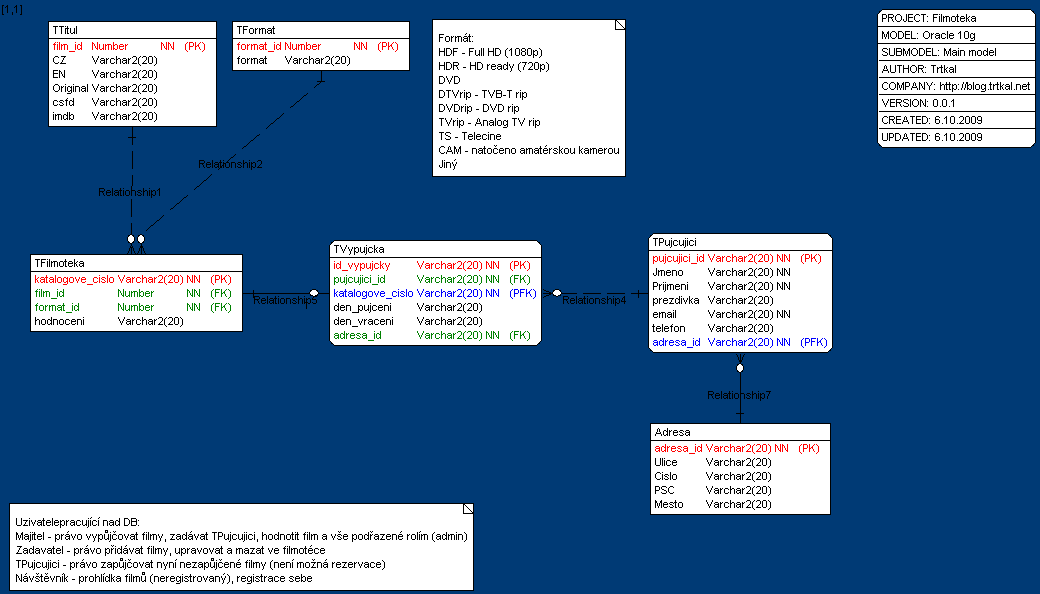
}

### Odpojení od databáze

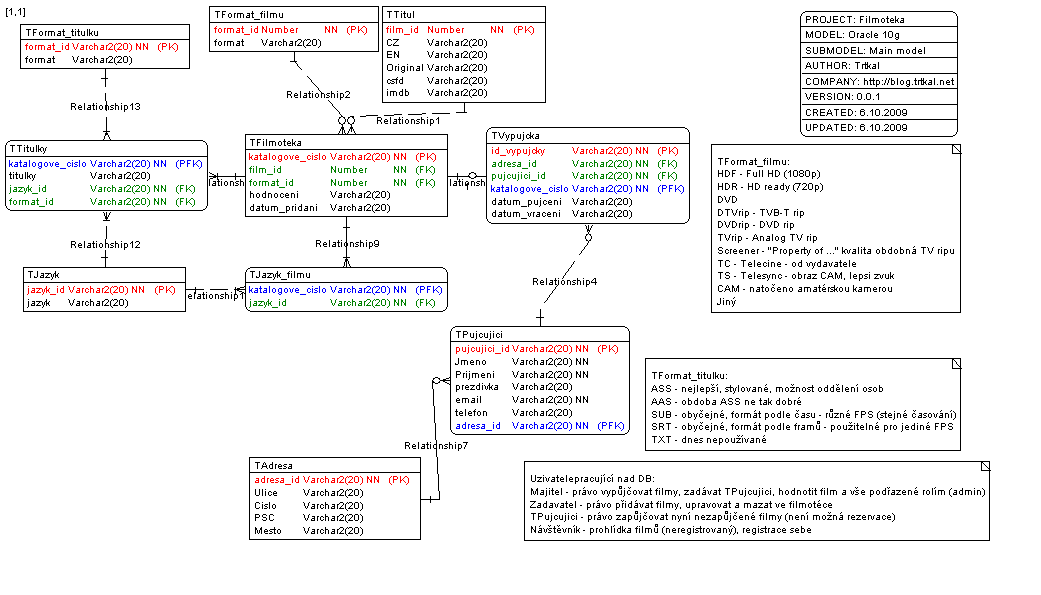
public function \_\_destruct() {  
 try {  
 dibi::disconnect();  
 } catch (DibiException $e) {  
 echo get\_class($e), ': ', $e->getMessage(), "\n";  
 }  
 }

# E-R Diagram

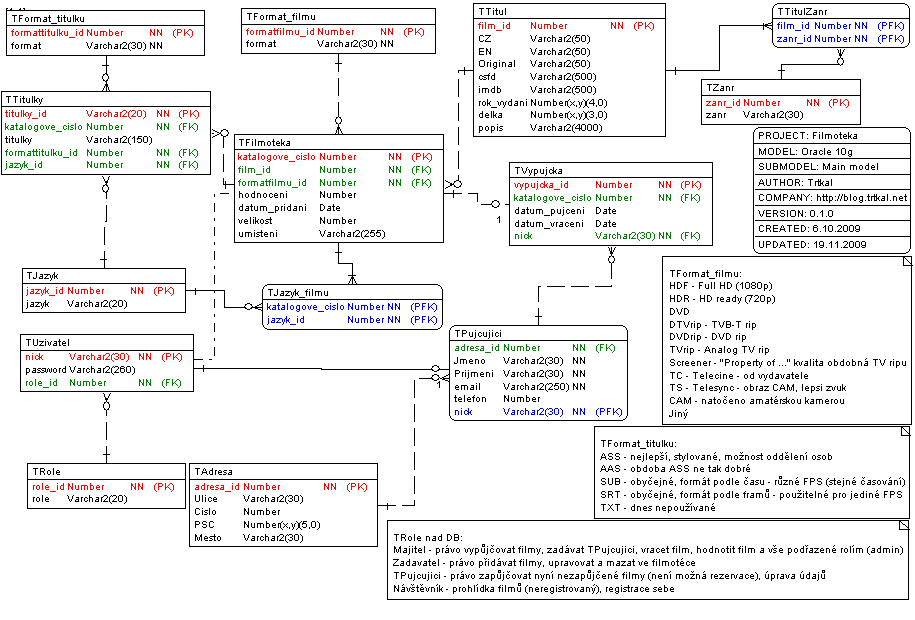
Základní návrh databáze Oracle se postupně vyvinul do značně většího modelu, než se kterým se podle zadání a návrhu počítalo.



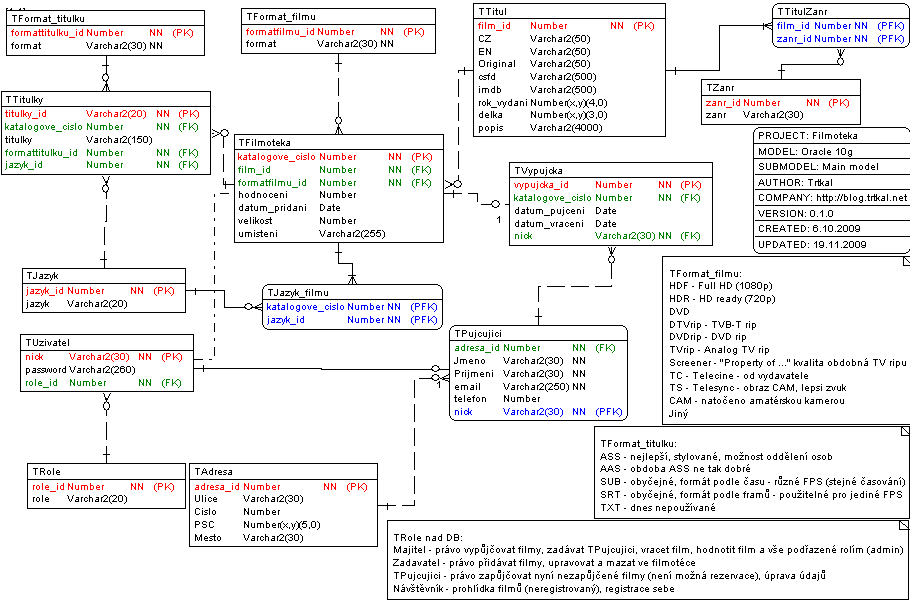
Obrázek Předpokládaný rozsah databáze



Obrázek Postup vývoje databázové struktury



Obrázek Téměř finální podoba struktury tabulek



Obrázek Finální podoba databáze

# Popis tabulek a jejich atributů

### TZanr

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| zanr\_id | Number [PK] | ID žánru filmu |
| Zanr | Number | Textový popis žánru |

### TTitulZanr

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| film\_id | Number [PFK] | ID žánru filmu |
| zanr\_id | Number [PFK] | ID titulu filmu |

### TFormat\_filmu

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| formatfilmu\_id | Number [PK] | ID formátu filmu |
| format | Varchar(30) [NN] | Textový popis formátu filmu |

### TFormat\_titulku

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| formattitulku\_id | Number [PK] | ID formátu titulků |
| format | Varchar(30) [NN] | Textový popis formátu titulků |

### TJazyk

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| jazyk\_id | Number [PK] | ID jazyku filmu a titulků |
| jazyk | Varchar(20) [NN] | Textový popis jazyku |

### TTitulky

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| titulky\_id | Number [PK] | ID titulků filmu |
| katalogove\_cislo | Number [FK] | Katalogové číslo filmu |
| titulky | Varchar2(150) | Název souboru titulků |
| formattitulku\_id | Number [FK] | ID formátu titulků |
| jazyk\_id | Number [FK] | ID jazyku titulků |

### TJazyk\_filmu

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| katalogove\_cislo | Number [PFK] | Katalogové číslo filmu |
| jazyk\_id | Number [PFK] | ID jazyku filmu |

### TRole

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| role\_id | Number [PK] | ID role uživatelů |
| Role | Varchar(20) [NN] | Textový popis role uživatelů |

### TUzivatel

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| Nick | Varchar(30) [PK] | Přezdívka uživatele |
| password | Varchar(260) [NN] | Heslo uživatele (šifrované pomocí SHA1 + MD5) |
| role\_id | Number [FK] | ID role uživatelů |

### TTitul

| Název atributu | Vlastnosti a typ | Popis |
| --- | --- | --- |
| film\_id | Number [PK] | ID titulu filmu |
| CZ | Varchar2(50) | Český název titulu |
| EN | Varchar2(50) | Anglický název titulu |
| Original | Varchar2(50) [NN] | Originální název titulu |
| Csfd | Varchar2(500) | Odkaz na filmovou databázi ČSFD.cz |
| Imdb | Varchar2(500) | Odkaz na filmovou databázi imdb.com |
| rok\_vydani | Number(4,0) [NN] | Rok vydání titulu |
| Delka | Number(3,0) [NN] | Délka titulu |
| Popis | Varchar2(4000) | Krátký popis titulu |

### TAdresa

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| adresa\_id | Number [PK] | ID adresy půjčujícího |
| Ulice | Varchar2(3 [NN] | Název ulice půjčujícího |
| Cislo | Number [NN] | Číslo popisné |
| PSC | Number(5,0) [NN] | PSČ |
| Město | Varchar2(30) | Město půjčujícího *(duplicitní údaj díky PSČ)* |

### TPujcujici

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| adresa\_id | Number [FK] | ID adresy půjčujícího |
| Jmeno | Varchar2(30) [NN] | Jméno půjčujícího |
| Prijmeni | Varchar2(30) [NN] | Příjmení půjčujícího |
| email | Varchar2(250) [NN] | Email půjčujícího |
| telefon | Number | Telefon půjčujícího |
| Nick | Varchar2(30) [PFK] | Přezdívka uživatele, který patří k půjčujícímu |

### TVypujcka

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| vypujcka\_id | Number [PK] | ID výpůjčky |
| katalogove\_cislo | Number [FK] | Katalogové číslo filmu |
| datum\_pujceni | Date [NN] | Datum vypůjčení filmu |
| datum\_vraceni | Date | Datum vrácení filmu |
| Nick | Varchar2(30) [FK] | Přezdívka uživatele, který patří k půjčujícímu |

### TFilmoteka

|  |  |  |
| --- | --- | --- |
| Název atributu | Vlastnosti a typ | Popis |
| katalagove\_cislo | Number [PK] | Katalogové číslo filmu |
| film\_id | Number [FK] | ID titulu filmu |
| formatfilmu\_id | Number [FK] | ID formátu filmu |
| hodnoceni | Number | Hodnocení filmu majitelem (hvězdičky) |
| datum\_pridani | Date [NN] | Datum přidání do filmotéky |
| velikost | Number | Velikost filmu na disku [MB] |
| umisteni | Varchar(255) | Volitelný údaj o tom, kde se film nachází |

# Popis a syntaxe použitých databázových objektů

## Funkce

Funkce vracející celkový počet nevrácených filmů.

***create or replace  
function*** *f\_pocetnevracenychfilmu* ***return number as*** *pocet\_pujcenych* ***number****;****begin******select******count****(\*)* ***into*** *pocet\_pujcenych* ***from*** *tpujcujici* ***left join*** *tvypujcka* ***on*** *tvypujcka.datum\_vraceni* ***is null******where*** *tvypujcka.nick* ***=*** *tpujcujici.nick* ***order by*** *tpujcujici.prijmeni;* ***return*** *pocet\_pujcenych;****end*** *f\_pocetnevracenychfilmu;****/****--* ***select*** *f\_pocetnevracenychfilmu* ***as*** *NEVRACENYCH* ***from*** *dual****;***

Vypíše celkový počet filmů v databázi.

***create or replace function*** *f\_pocetTitulu****return number as*** *v\_pocet number;****begin******select count****(\*)* ***into*** *v\_pocet* ***from*** *ttitul;* ***return*** *v\_pocet;****end****;****/****--* ***select*** *f\_pocetTitulu* ***as*** *POCETTITULU* ***from*** *dual****;***

## Trigery

Spouštěče pro automatické číslování primárních klíčů v tabulkách.

***create or replace trigger*** *tformattitulku\_id* ***before insert on*** *tformat\_titulku* ***for each row begin select*** *sformattitulku\_id.nextval* ***into*** *:new.formattitulku\_id* ***from*** *dual; end;****create or replace trigger*** *ttitulky\_id* ***before insert on*** *ttitulky* ***for each row begin select*** *stitulky\_id.nextval* ***into*** *:new.titulky\_id* ***from*** *dual; end;****create or replace trigger*** *tjazyk\_id* ***before insert on*** *tjazyk* ***for each row begin select*** *sjazyk\_id.nextval* ***into*** *:new.jazyk\_id* ***from*** *dual; end;****create or replace trigger*** *trole\_id* ***before insert on*** *trole* ***for each row begin*** *select srole\_id.nextval into :new.role\_id* ***from*** *dual; end;****create or replace trigger*** *tadresa\_id* ***before insert on*** *tadresa* ***for each row begin select*** *sadresa\_id.nextval* ***into*** *:new.adresa\_id* ***from*** *dual; end;****create or replace trigger*** *tvypujcka\_id* ***before insert on*** *tvypujcka* ***for each row begin select*** *svypujcka\_id.nextval* ***into*** *:new.vypujcka\_id* ***from*** *dual; end;****create or replace trigger*** *tfilm\_id* ***before insert on******ttitul for each row begin select*** *sfilm\_id.nextval* ***into*** *:new.film\_id* ***from*** *dual; end;****create or replace trigger*** *tzanr\_id* ***before insert on*** *tzanr* ***for each row begin select*** *szanr\_id.nextval* ***into*** *:new.zanr\_id* ***from*** *dual; end;****create or replace trigger*** *tkatalogove\_****cislo before insert on*** *tfilmoteka* ***for each row begin select*** *skatalogove\_cislo.nextval* ***into*** *:new.katalogove\_cislo* ***from*** *dual; end;****create or replace trigger*** *tformatfilmu\_id* ***before insert on*** *tformat\_filmu* ***for each row begin******select*** *sformatfilmu\_id.nextval* ***into*** *:new.formatfilmu\_id* ***from*** *dual; end;****/***

## Indexy

***CREATE INDEX*** *I\_FILM\_ID* ***ON*** *TTITUL (FILM\_ID)   
/****CREATE******INDEX*** *I\_KATALOGOVE\_CISLO* ***ON*** *TFILMOTEKA (KATALOGOVE\_CISLO)   
/****CREATE******INDEX*** *I\_VYPUJCKA\_ID* ***ON*** *TVYPUJCKA (VYPUJCKA\_ID)   
/****CREATE******INDEX*** *I\_TITULKY\_ID* ***ON*** *TTITULKY (TITULKY\_ID)   
/*

## Sekvence

Sekvence pro indexování tabulek – autoincrement.

**CREATE SEQUENCE** sformattitulku\_id **INCREMENT BY 1 START WITH 1 nomaxvalue;**  
**CREATE SEQUENCE** stitulky\_id **INCREMENT BY 1 START WITH 1 nomaxvalue;  
CREATE SEQUENCE** sjazyk\_id **INCREMENT BY 1 START WITH 1 nomaxvalue;**  
**CREATE SEQUENCE** srole\_id **INCREMENT BY 1 START WITH 1 nomaxvalue;  
CREATE SEQUENCE** sadresa\_id **INCREMENT BY 1 START WITH 1 nomaxvalue;  
CREATE SEQUENCE** svypujcka\_id **INCREMENT BY 1 START WITH 1 nomaxvalue;  
CREATE SEQUENCE** sfilm\_id **INCREMENT BY 1 START WITH 1 nomaxvalue;  
CREATE SEQUENCE** szanr\_**id INCREMENT BY 1 START WITH 1 nomaxvalue;  
CREATE SEQUENCE** skatalogove\_cislo **INCREMENT BY 1 START WITH 1 nomaxvalue;  
CREATE SEQUENCE** sformatfilmu\_id **INCREMENT BY 1 START WITH 1 nomaxvalue;  
/**

## Pohledy

Pohled pro zobrazení počtu titulů v databázi. Vhodnější by bylo použití funkce.

**CREATE OR REPLACE FORCE VIEW** movie\_count ("POCET\_FILMU") **AS SELECT** **count**(film\_id) **as** pocet\_filmu **FROM** ttitul;

Pohled pro zobrazení počtu uživatelů v databázi. Vhodnější by bylo použití funkce,

**CREATE OR REPLACE FORCE VIEW** users\_count ("POCET\_UZIVATELU") **AS SELECT** **count**(nick) **as** pocet\_uzivatelu **FROM** tuzivatel;

## Defaultní předpřipravená data pro fungování webové části

/\* Přístupové role \*/  
**INSERT INTO** "ST22312"."TROLE" (ROLE) **VALUES** ('Visitor');  
**INSERT INTO** "ST22312"."TROLE" (ROLE) **VALUES** ('Borrower');  
**INSERT INTO** "ST22312"."TROLE" (ROLE) **VALUES** ('Administrator');

Pro získání hesla je třeba použít skriptu:

<?php  
 $password = “nekriptované heslo“;  
 echo sha1($password) . md5(strlen($password));  
?>

/\* Vytvoření základního administrátora s přístupem do administrace \*/  
***INSERT INTO*** *TUZIVATEL (NICK,PASSWORD,ROLE\_ID)* ***VALUES*** *('Admin','kriptované heslo',3);*

## Další operace nad databází

/\* zjištění nevrácených výpůjček \*/  
**select** tpujcujici.jmeno **as** JMENO, tpujcujici.prijmeni **as** PRIJMENI, **to\_date**(tvypujcka.datum\_pujceni, 'DD.MM.RR') **as** DATUM\_PUJCENI, ttitul.cz **as** CZ, ttitul.original **as** ORIGINAL **from** tpujcujici, tfilmoteka, ttitul  
**left join** tvypujcka **on** tvypujcka.datum\_vraceni **is null**  
**where** tvypujcka.nick = tpujcujici.nick **and** tvypujcka.katalogove\_cislo= tfilmoteka.katalogove\_cislo **and** tfilmoteka.film\_id=ttitul.film\_id  
**order** **by** tpujcujici.prijmeni;

/\* přidání uživatele \*/  
**insert** **into** tuzivatel (nick, password, role\_id) **values** ('".$username."','".$password."','".$role."')

/\* Automatické doplňování hledaného textu v databázi \*/  
**select** film\_id **as** IDFILMU, cz **as** CZ, en **as** EN, original **as** ORIGINAL, **to\_char**(**substr**(popis,0,50)) **as** POPIS **from** ttitul **where** **lower**(ttitul.cz) **like** **lower**('%".$var."%') **or** **lower**(ttitul.en) **like** **lower**('%".$var."%') **or** **lower**(ttitul.original) **like** **lower**('%".$var."%')

/\* počet nevrácených filmů \*/  
**select count**(\*) **from** tpujcujici  
**left join** tvypujcka **on** tvypujcka.datum\_vraceni **is null**where tvypujcka.nick = tpujcujici.nick  
**order by** tpujcujici.prijmeni;

/\* Ověření uživatele proti databázi \*/  
**select** role **as** ROLE, nick **as** USERNAME, password **as** PASSWORD **from** TUZIVATEL **left join** TROLE **on** TROLE.ROLE\_ID=TUZIVATEL.ROLE\_ID **where** NICK='".$username."' **and** PASSWORD='".$password."'

# Závěr

Na této semestrální práci jsem si procvičil, oživil a naučil lépe využívat možnosti databázového serveru. Asi největší problém bylo navrhnout tabulky, zajistit konzistenci a propojení dat. Další problém nastal, při propojování databáze s PHP, jelikož jsem využil služeb Dibi se kterou jsem doposud nikdy nepracoval. Semestrální práce mi dala bohaté nově nabité znalosti pro další vývoj aplikací.

# Seznam obrázků

[Obrázek 1 UML Use case diagram 4](#_Toc248596615)

[Obrázek 2 Předpokládaný rozsah databáze 8](#_Toc248596616)

[Obrázek 3 Postup vývoje databázové struktury 8](#_Toc248596617)

[Obrázek 4 Téměř finální podoba struktury tabulek 9](#_Toc248596618)

[Obrázek 5 Finální podoba databáze 9](#_Toc248596619)